**在运行时动态指定谓词筛选器（C# 编程指南）**

在某些情况下，您直到运行时才能知道必须将多少个谓词应用于 **where** 子句中的源元素。动态指定多个谓词筛选器的一种方式是使用 [Contains<TSource>](https://msdn.microsoft.com/zh-cn/library/bb352880.aspx) 方法，如下面的示例中所示。此示例按两种方式构造。首先，通过筛选程序中提供的值来运行项目。然后，使用运行时提供的输入再次运行项目。

**使用 Contains 方法进行筛选**

1. 在 Visual Studio 中，打开一个新的控制台应用程序。将其命名为 PredicateFilters。
2. 从[如何：查询对象集合（C# 编程指南）](https://msdn.microsoft.com/zh-cn/library/bb907066.aspx)中复制 StudentClass 类，并将此类粘贴到 Program 类下方的命名空间 PredicateFilters 中。 StudentClass 提供了 Student 对象的列表。
3. 将 StudentClass 中的 Main 方法注释掉。
4. 将 Program 类替换为以下代码。

class DynamicPredicates : StudentClass

{

static void Main(string[] args)

{

string[] ids = { "111", "114", "112" };

Console.WriteLine("Press any key to exit.");

Console.ReadKey();

}

static void QueryByID(string[] ids)

{

var queryNames =

from student in students

let i = student.ID.ToString()

where ids.Contains(i)

select new { student.LastName, student.ID };

foreach (var name in queryNames)

{

Console.WriteLine("{0}: {1}", name.LastName, name.ID);

}

}

}

1. 在 DynamicPredicates 类的 Main 方法中的 ids 声明下添加以下行。
2. QueryById(ids);
3. 按 F5 运行项目。
4. 命令提示符窗口中将显示以下输出：

Garcia: 114

O'Donnell: 112

Omelchenko: 111

1. 下一步是重新运行项目，这次使用在运行时输入的输入内容而不是数组 ids 来运行项目。在**“解决方案资源管理器”**中右击**“PredicateFilters”**，然后单击**“属性”**。
2. 单击**“调试”**选项卡。
3. 在**“命令行参数”**窗口中，按如下方式键入 122、117、120 和 115（用空格隔开）：**122 117 120 115**。在运行项目时，这些值将成为 Main 方法的参数 args 的元素。
4. 在 Main 方法中，将 QueryByID(ids) 更改为 QueryByID(args)。
5. 按 F5 运行项目。
6. 命令提示符窗口中将显示以下输出：

Adams: 120

Feng: 117

Garcia: 115

Tucker: 122

**使用 switch 语句进行筛选**

1. 可以使用 **switch** 语句在预先确定的备选查询中进行选择。在下面的示例中，studentQuery 使用的 **where** 子句随着在运行时指定的年级或年份的不同而不同。
2. 复制以下方法并将其粘贴到类 DynamicPredicates 中。

// To run this sample, first specify an integer value of 1 to 4 for the command

// line. This number will be converted to a GradeLevel value that specifies which

// set of students to query.

// Call the method: QueryByYear(args[0]);

static void QueryByYear(string level)

{

GradeLevel year = (GradeLevel)Convert.ToInt32(level);

IEnumerable<Student> studentQuery = null;

switch (year)

{

case GradeLevel.FirstYear:

studentQuery = from student in students

where student.Year == GradeLevel.FirstYear

select student;

break;

case GradeLevel.SecondYear:

studentQuery = from student in students

where student.Year == GradeLevel.SecondYear

select student;

break;

case GradeLevel.ThirdYear:

studentQuery = from student in students

where student.Year == GradeLevel.ThirdYear

select student;

break;

case GradeLevel.FourthYear:

studentQuery = from student in students

where student.Year == GradeLevel.FourthYear

select student;

break;

default:

break;

}

Console.WriteLine("The following students are at level {0}", year.ToString());

foreach (Student name in studentQuery)

{

Console.WriteLine("{0}: {1}", name.LastName, name.ID);

}

}

1. 在**“命令行参数”**窗口中，将上一过程中的 ID 号替换为一个介于 1 到 4 之间的整数值。
2. 在 Main 方法中，将对 QueryByID 的调用替换为以下调用，此调用可将 args 数组中的第一个元素作为其参数发送：QueryByYear(args[0])。
3. 按 F5 运行项目。

**在您自己的应用程序中使用此方法**

* 在改编此方法以适合您自己的应用程序时，请记住，LINQ 需要版本 3.5 或版本 4 的 .NET Framework，并且项目必须包含一个对 System.Core.dll 的引用和一条针对 **System.Linq** 的 **using** 指令。LINQ to SQL、LINQ to XML 和 LINQ to DataSet 类型需要相应的其他 **using** 指令和引用。有关更多信息，请参见 [How to: Create a LINQ Project](https://msdn.microsoft.com/zh-cn/library/bb546156.aspx)。